

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

**Red-Eye Detection Based On Red Region Detection
With Eye Confirmation**

Inventor(s):

Tong-Xian Chen

Xiangrong Chen

John Platt

Jie Yan

Hong-Jiang Zhang

ATTORNEY'S DOCKET NO. MS1-724US

1 **TECHNICAL FIELD**

2 This invention relates to detecting red eye, and more particularly to red-eye
3 detection based on red region detection with eye confirmation.
4

5 **BACKGROUND OF THE INVENTION**

6 Red-eye is a problem commonly encountered in photography when light
7 (typically from the camera's flash) reflects off the retinas at the back of the
8 subject's eyes and causes the subject's eyes to turn red. Red-eye has been a
9 problem for many years, and although a variety of solutions have been proposed to
10 cure the problem, these solutions tend to be costly, cumbersome, and/or
11 ineffective. One such solution is to use a bounce flash so that light hits the
12 subject's eyes from the side (or above/below) rather than straight-on, thereby
13 preventing the reflected light from coming straight back to the camera's lens.
14 Bounce flashes, however, are cumbersome (often rivaling the size of the camera)
15 and costly. Another solution is to pre-flash the subject, thereby causing the
16 subject's pupils to close and decrease the amount of light allowed into the subject's
17 eyes when the picture is taken. These pre-flash solutions, however, are not always
18 effective, and cause a delay (while the pre-flash is operating) before the picture is
19 actually taken during which time the subject may move.

20 Attempts have also been made to cure the red-eye problem after-the-fact by
21 processing the image to remove the red from the eyes. Computer software
22 packages are available that allow for the removal of red-eye, such as by changing
23 the color of the red portion of the eye. Some systems require manual selection, by
24 the user, of the pixels within the image that are part of the red eyes prior to
25

1 removing the red-eye. These systems are rather user un-friendly due to the steps
2 the user must follow to identify exactly which pixels are part of the red eyes.

3 Other systems have attempted to automatically detect where the red-eye
4 portions of an image are (as opposed to other non-eye portions of the image that
5 are red). Such systems typically start by using face detection techniques to
6 determine where any faces are in the image and where eyes are within those faces.
7 Once these faces (and eyes within them) are detected, the systems try to determine
8 whether the eyes are red eyes. These systems, however, can have poor
9 performance under many circumstances (e.g., when a face is partially obscured,
10 such as by heavy shadows or heavy beards, when the face has an unusual
11 expression or is distorted, etc.).

12 The invention described below addresses these disadvantages, providing
13 improved red-eye detection systems and methods.

14 **SUMMARY OF THE INVENTION**

15 Red-eye detection based on red region detection with eye confirmation is
16 described herein.

17 In accordance with one aspect, pixels that correspond to the color of red-
18 eye within an image are identified. A determination is then made as to whether
19 these identified pixels and surrounding areas are part of an eye or not part of an
20 eye. Those identified pixels that are determined to be part of an eye are the
21 detected red-eye regions.

22 In accordance with another aspect, a skin color filter is initially applied to a
23 received image to identify areas of the image that include skin color. Those areas
24 are then searched to identify red pixels within the areas. Adjacent red pixels (or
25

those red pixels close enough to one another) are grouped together and a shape filter applies several rules to the pixel groupings. Pixel groups remaining after the filtration process are candidate red-eye regions. These candidate red-eye regions are input to an eye confirmation process which uses a multi-scale process to confirm whether each candidate red-eye region is part of an eye.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings. The same numbers are used throughout the figures to reference like components and/or features.

Fig. 1 illustrates an exemplary environment in which the present invention may be practiced.

Fig. 2 illustrates an exemplary system for performing red-eye detection in accordance with certain embodiments of the invention.

Fig. 3 illustrates an example of scaling an image in accordance with certain embodiments of the invention

Fig. 4 is a flowchart illustrating an exemplary process for detecting red-eye regions in accordance with certain embodiments of the invention.

Fig. 5 illustrates an example of a suitable operating environment in which the invention may be implemented.

DETAILED DESCRIPTION

Fig. 1 illustrates an exemplary environment in which the present invention may be practiced. A camera 102 is used to take a picture(s) of a subject 104 using a flash on camera 102 that potentially causes red-eye in images captured of subject

1 104. These images captured by camera 102 (commonly referred to as a picture
2 being taken) are analyzed for red-eye and the areas with red-eye automatically
3 detected as discussed in more detail below. The red-eye detection can be
4 performed at camera 102, or alternatively the captured images may be transferred
5 to a computing device 106 that detects red-eye. Device 106 may be any of a wide
6 variety of devices, such as a desktop or portable computer, copying or printing
7 devices (e.g., a photograph enlargement device including a scanner and printer),
8 etc.

9 Camera 102 may be any type of image capture device that captures and
10 stores (or communicates) images, such as a film camera, a digital camera, a video
11 camera, a camcorder, etc. Camera 102 may capture images in any of a variety of
12 conventional manners, such as exposing film on which the image is captured,
13 exposing one or more charge coupled devices (CCDs) and storing a captured still
14 image in memory (e.g., a removable Flash memory, hard disk (or other magnetic
15 or optical storage medium), or motion video tape), exposing one or more CCDs
16 and storing multiple captured frames (a captured video sequence) on a recording
17 medium (e.g., Flash memory, disk or tape), etc.

18 Fig. 2 illustrates an exemplary system 120 for performing red-eye detection
19 in accordance with certain embodiments of the invention. System 120 can be
20 implemented in any of a wide variety of devices, such as computers (whether
21 desktop, portable, handheld, etc), image capture devices (e.g., camera 102 of Fig.
22 1), etc. Alternatively, system 120 may be a standalone system for coupling to (or
23 incorporation within) other devices or systems.

24 System 120 receives an image 122 into a red region detection module 124.
25 Image 122 is received in digital format, but can be received from any of a wide

1 variety of sources including sources that capture images in a non-digital format
2 (e.g., on film) but that are subsequently converted to digital format (digitized). In
3 the illustrated example, image 122 is made up of multiple pixels that can be
4 referenced in a conventional manner using an x, y coordinate system. Red region
5 detection module 124 detects red regions that are potentially regions of red-eye
6 and identifies those detected regions to an eye confirmation module 126. Eye
7 confirmation module 126 confirms each detected region as being either part of an
8 eye or not part of an eye, and outputs an indication 128 of those detected regions
9 that are confirmed as being parts of eyes. The identified detected red-eye regions
10 128 can then be made available to other systems for further processing, such as
11 automatic removal of the red-eye regions (e.g., by changing the red color to
12 black).

13 Red region detection module 124 includes a red pixel identifier 130, a pixel
14 grouper 132, and a filter 134. Image 122 is received by red pixel identifier 130
15 which analyzes image 122 on a per-pixel basis and identifies which of the pixels
16 are "red" pixels. These identified red pixels are those pixels having a color that is
17 associated with the colors typically found in red-eye. Identifier 130 may analyze
18 each pixel in image 122, or alternatively only a subset of the pixels in image 122.
19 For example, if a large number of red pixels in a circular pattern are identified then
20 some of the pixels in the center of that pattern need not be analyzed. By way of
21 another example, analysis of some other pixels may simply be skipped (e.g., at the
22 corners or edges of the image), although skipping such analysis may degrade the
23 performance of the red-eye detection.

24 In one implementation, skin color filter module 136 detects those areas of
25 image 122 that include skin color and communicates those images to identifier

1 130, thereby allowing identifier 130 to analyze only those pixels that are within
2 the areas that include skin color. Different skin color filters can be applied by
3 module 136, and in one implementation a skin color classifier is used in which
4 color quantization of the original image is initially performed in order to improve
5 skin color segmentation by homogenizing the image regions. The quantized color
6 image is then segmented according to skin color characteristics based on either the
7 YCbCr color model or the HSV (Hue, Saturation, Value) color model. This color
8 quantization and image segmentation is discussed in more detail in Christophe
9 Garcia and Georgios Tziritas, "Face Detection Using Quantized Skin Color
10 Regions Merging and Wavelet Packet Analysis", IEEE Transactions on
11 Multimedia, Vol. 1, No. 3, September 1999, which is hereby incorporated by
12 reference.

13 Given that red-eye is not typically a single shade of red, pixel identifier 130
14 uses a red-eye color model to which the color of each pixel being analyzed is
15 compared. Based on this comparison to the red-eye color model, pixel identifier
16 130 determines whether the pixel is or is not a red pixel.

17 In the illustrated example, the pixels of image 122 are 24-bit color pixels
18 that are represented using the conventional RGB (Red, Green, Blue) color model,
19 in which three different dots (one red, one green, and one blue) are energized to
20 different intensities to create the appropriate color for the pixel. The 24 bits of
21 color information identify the intensity that each of the three different dots is to be
22 energized to in order to display the pixel. The RGB color model is well known to
23 those skilled in the art and thus will not be discussed further except as it pertains to
24 the present invention.
25

1 Identifier 130 converts the 24-bit color model using RGB into a two-
2 dimensional space referred to herein as the g and γ characteristics. The g and γ
3 characteristics are determined based on the three components of the RGB model as
4 follows:

$$5 \quad g = \frac{G}{R + G + B}$$

$$7 \quad \gamma = \frac{R}{R + G + B}$$

8 This two-dimensional space using the g and γ characteristics is previously
9 trained (e.g., offline) using multiple color samples from known red-eye pixels.
10 This results in a two-dimensional Gaussian distribution for red-eye colors based
11 on the g and γ characteristics. Once the g and γ characteristics are generated for
12 the pixel being analyzed, the g and γ characteristics for that pixel are compared to
13 the Gaussian distribution. If the g and γ characteristics of the pixel are within a
14 threshold probability of the Mixture Gaussian distribution, then identifier 130
15 determines that the pixel is a red pixel; otherwise identifier 130 determines that the
16 pixel is not a red pixel. In one implementation, the threshold probability is 0.6,
17 although different values could alternatively be used.

18 Identifier 130 outputs an identification of each of the red pixels to pixel
19 grouper 132. This identification can take any of a variety of forms, such as an x, y
20 coordinate position of each pixel. Each of the identified pixels is a candidate red-
21 eye pixel. Pixel grouper 132 groups together the candidate red-eye pixels
22 identified by identifier 130 into one or more pixel groups. Pixel grouper 132
23 groups together any two adjacent candidate red-eye pixels into the same pixel
24 group. In one implementation, two pixels are adjacent if each of their x and y
25

1 coordinate values differs by no greater than one. Thus, each pixel surrounding a
2 given pixel (whether above, below, to the left, to the right, or at a diagonal) is an
3 adjacent pixel. Alternatively, surrounding diagonal pixels may not be considered
4 adjacent. Pixel grouper 132 may optionally group together two candidate red-eye
5 pixels that are not adjacent but are within a threshold distance of one another (e.g.,
6 separated by not more than one or two pixels) into the same group. Pixel grouper
7 132 then identifies these pixel groups (which includes any single pixels as their
8 own groups) to filter 134.

9 Filter 134 filters the pixel groups received from pixel grouper 132 based on
10 a set of one or more heuristic rules. Filter 134 identifies certain pixel groups as
11 being potential (or candidate) red-eye regions based on these rules and outputs an
12 identification of the candidate red-eye regions to eye confirmation module 126.

13 In one implementation, filter 134 begins by assuming all of the pixel groups
14 received from grouper 132 are candidate red-eye regions, and then uses the
15 following five rules to eliminate pixel groups. The pixel groups remaining (if any)
16 after application of these rules are the candidate red-eye regions output to eye
17 confirmation module 126.

18 Rule 1) If the entire image is red (e.g., greater than a threshold number
19 amount of the pixels in the picture are red, such as 95%) then none of the pixel
20 groups are red-eye regions.

21 Rule 2) A pixel group containing too few pixels (e.g., five or less) is not a
22 red-eye region.

23 Rule 3) A pixel group that is more rectangular than circular is not a red-eye
24 region. The shape of a pixel group can be determined in any of a wide variety of
25 conventional manners, such as based on the circumference of the group.

1 Rule 4) A pixel group having an aspect ratio substantially different from a
2 circle is not a red-eye region. The aspect ratio of the pixel group can be identified
3 by calculating the distance between the rightmost and leftmost pixels in the group
4 (the horizontal aspect), as well as the distance between the uppermost and
5 lowermost pixels in the group (the vertical aspect). The aspect ratio is then the
6 horizontal aspect divided by the vertical aspect. For a circle, the aspect ratio is
7 one. In the illustrated example, a pixel group with an aspect ratio less than a lower
8 bound or greater than an upper bound is not a red-eye region. In one
9 implementation, the lower bound is 0.5 and the upper bound 2.0.

10 Rule 5) A pixel group having a low filling ratio (e.g., less than 0.6) is not a
11 red-eye region. The filling ratio is the number of red pixels in the group divided
12 by the product of the horizontal aspect and the vertical aspect (as described in Rule
13 4).

14 Filter 134 identifies the resultant candidate red-eye regions to eye
15 confirmation module 126. This identification can take any of a variety of forms,
16 such as the x , y coordinates of each pixel in the grouping, one pixel of the
17 grouping (e.g., at the center) and the size of the grouping, a single pixel of the
18 grouping (e.g., at the center), etc. Eye confirmation module 126 moves a window
19 the size of an eye template (also referred to as an eye detector) around the image
20 (e.g., starting with the eye template at or close to the center of the grouping) and
21 determines whether the pixels of the image within the windows match the eye
22 template. The eye template is trained based on multiple previously analyzed (e.g.,
23 offline) eyes, which includes both the pupil areas (which include the red-eye
24 portion) and the areas surrounding the pupil (which may include, for example, the
25 iris and the sclera, as well as possibly the skin, eyelashes, and eyebrows

1 surrounding the eyeball, etc.). The eye confirmation module 126 analyzes the area
2 surrounding the candidate red-eye regions to determine whether the regions are
3 part of an eye (and thus truly red-eye regions) or not part of an eye (and thus not
4 red-eye regions). However, only areas close to the pupil are analyzed (the entire
5 face is not detected). In one implementation the window is 25 pixels
6 (horizontally) by 15 pixels (vertically), although windows of other sizes may be
7 used.

8 To perform the confirmation for a particular candidate red-eye region, the
9 window is positioned over (e.g., centered on) the red-eye region and the pixels
10 within the window are classified, based on the eye template, as being either an eye
11 or not an eye. If the pixels are classified as an eye, then no further analysis need
12 be made for that eye. Alternatively, additional analysis may be performed (by
13 moving the window over the image in the horizontal and/or vertical directions and
14 repeating the classification) in order to identify the actual location (boundaries) of
15 the eye (e.g., the eyeball, including the pupil, iris, and sclera). However, if the
16 pixels are classified as not an eye, then the window is adjusted in the horizontal
17 and/or vertical direction and the classification repeated. The window can be
18 moved around multiple times and in multiple directions in an attempt to "locate"
19 the eye if the candidate red-eye region is indeed a red-eye region. In one
20 implementation, movement of the window is limited to ranging from -3 pixels to
21 +3 pixels from the starting location in both the horizontal and vertical directions.

22 The classification of the pixels within the window based on the trained eye
23 template can be performed using any of a wide variety of conventional
24 classification schemes. In one implementation, a conventional SVM (Support
25 Vector Machine) classifier is trained using both images of eyes and images of

1 objects similar to eyes but that are not eyes. Based on this training, the SVM
2 classifier can classify the pixels in a window as being either an eye or not an eye.
3 In another implementation, the pixels in the window are normalized to account for
4 variations in lighting conditions and then projected onto an eigenspace
5 representation which returns a feature vector for the candidate eye region. This
6 feature vector is then input to a neural network trained with images of eyes, which
7 classifies the feature vector as either an eye or not an eye. The operation and use
8 of SVM and neural networks for classification are both well-known to those
9 skilled in the art and thus will not be discussed further except as they pertain to the
10 present invention.

11 In the illustrated example, the size of the eye template and the size of the
12 window are both fixed. Thus, some accommodation is made to account for the
13 different sizes of eyes that may appear in images (e.g., based on how close the
14 camera is to the subject, the size of the subject, how much the camera may have
15 been "zoomed" for the picture, etc.). In one implementation, rather than having a
16 fixed size eye template and window, multiple different-sized eye templates and
17 windows are used to accommodate for these differences.

18 In another implementation, the eye template and window sizes remain
19 fixed, but the scale of the image is modified. Fig. 3 illustrates an example of the
20 scaling of the image in accordance with certain embodiments of the invention. An
21 image 160 is illustrated including multiple candidate red-eye regions 162, 164,
22 166, 168, and 170. For ease of illustration and to avoid cluttering the drawings,
23 only the candidate red-eye regions are illustrated (other elements of the picture are
24 not shown). Assume that the candidate red-eye regions 162 and 164 are actually
25 red-eye regions, but that the regions 166, 168, and 170 are not. A window 172 is

1 shown that covers some of the pixels of region 162, but not all of the region. This
2 remains true regardless of how window 172 is moved over image 160. In other
3 words, the region 162 is larger than the window size 172 (e.g., the camera may
4 have been very close to the subject when the picture was taken). Thus, it is
5 unlikely that the classification process will identify region 162 as a red-eye region.

6 As part of the multiple-scale process, image 160 is "scaled down" by 50%
7 to generate image 174. Scaling down the image by a certain amount reduces the
8 number of pixels by that amount, which may be accomplished by combining
9 pixels. Such scaling of an image can be performed in any of a wide variety of
10 conventional manners. The window 172 remains the same size, but the region
11 162 is reduced in size by 50%. The size of window 172 is now greater than the
12 size of region 162, but not by much. It is still possible that the classification
13 process may not identify region 162 as a red-eye region. Thus, image 172 is also
14 scaled down by 50%, resulting in image 176. Window 172 is now a good deal
15 larger than region 162, and it is likely that the classification process will now
16 identify region 162 as a red-eye region.

17 Although using a large number of scaling levels is good for red-region
18 confirmation, it may also become time-consuming. Thus, in order to reduce the
19 time required, in one implementation three scaling levels are used. The initial
20 scale is determined by the red region size, that is, the average of the size of the red
21 region's horizontal aspect and vertical aspect. For example, for a 10-pixel size red
22 region the initial scale is 1, while for a 20-pixel size red region the initial scale is
23 0.5 (reduced in size by 50%), etc. The second scale is the image from the initial
24 scale reduced in size by 5/6 (0.8333), and the third scale is the image from the
25 second scale reduced in size by another 5/6 (0.8333).

Returning to Fig. 2, eye confirmation module 126 outputs, as detected red-eye regions 128, each of the red-eye candidate regions that are confirmed as being eyes. In one implementation, the detected red-eye regions 128 are identified by the center of the eye and the size of the eye (e.g., in pixels) that includes the red-eye region. Alternatively, other identifications may be used, such as an identification of each of the red pixels within the pixel group that makes up the red-eye region.

In the discussions above, reference is made to "red-eye" detection. It is to be appreciated that this is typically various shades of the color red in images that people view, but may be different colors in different media. For example, conventional film cameras capture images on film which is then processed or developed resulting in what is typically referred to as a "negative". A printing process can then be used to print a color picture (or digitally capture a color picture) from this negative. However, the colors that constitute "red-eye" on a negative are not red. Nonetheless, the invention can be used to detect red-eye on negatives by re-training red pixel identifier 130 of Fig. 2 to detect the appropriate negative colors for red-eye. The negative image can then be digitized and the resultant digital image analyzed for red-eye.

Similarly, different animals may have red-eye that is in slightly different shades of red than humans. Such red-eye can also be detected by the invention by re-training red pixel identifier 130 of Fig. 2 to detect the appropriate shades for red-eye in the animal(s) being photographed. Similarly, different animals will have different eye shapes, and these too can be accounted for with the present invention by re-training the classifier used by eye confirmation module 126 of Fig. 2.

1 Additionally, other information may be incorporated into the red-eye
2 detection process. For example, information regarding whether a flash was used
3 in capturing the image may be used as part of the detection process – if no flash
4 was used then the red-eye detection process can simply determine that no red-eye
5 exists in the image. This information may be available to the red-eye detection
6 process by the camera (e.g., if the red-eye detection system is implemented in a
7 camera), or flash information may be stored along with the captured image (e.g.,
8 on film, or accompanying a digital image), etc.

9 In one implementation, captured images are stored digitally using a data
10 structure with a corresponding header. One example of such a header (often used
11 with JPEG-compressed files) is the Exchangeable Image File format (Exif),
12 including current and future versions of the format (additional information on the
13 Exif format is available from the Japan Electronic Industry Development
14 Association). This header information includes an indication as to whether a flash
15 was used in capturing the image. For digitally captured images, the digital camera
16 that captures the images will set or clear a flag within this header (the "Flash" tag
17 in Exif) for each image to indicate whether the camera's flash was used for
18 capturing each of the images. The red-eye detection process described herein can
19 check the header information for an image and, if the header information indicates
20 that no flash was used, then the process can quickly determine that no red-eye
21 exists without requiring further analysis of the image. Such quick determinations
22 can decrease the time required to process images, as well as reduce false
23 indications of red-eye. Alternatively, the header may also including information
24 (e.g., within the "Flash" tag in Exif) as to whether return light from the flash was
25 detected at the camera. If no return light was detected, then the red-eye detection

1 process may assume that there was insufficient flash light for the picture and thus
2 insufficient flash light for red-eye to result (and thus quickly determine that no
3 red-eye exists without requiring further analysis of the image).

4 Fig. 4 is a flowchart illustrating an exemplary process for detecting red-eye
5 regions in accordance with certain embodiments of the invention. The process of
6 Fig. 4 is carried out by system 120 of Fig. 2, and may be implemented in software.

7 Initially, an image is received (act 198). Optionally, a check is then made
8 as to whether a flash was used to capture the image (act 200). If no flash was used
9 (or no return flash light detected) then an indication that the image contains no
10 red-eye (act 202) and the process ends. However, if a flash was used, then red-eye
11 may exist within the image. Skin color is optionally used to identify areas to be
12 searched (act 204), after which the image is searched for candidate red pixels (act
13 206). If skin color is used to identify areas to be searched, then only those areas
14 identified in act 204 are searched in act 206; otherwise, all (or most) of the pixels
15 are searched in act 206.

16 The candidate red pixels that are adjacent (or close) to one another are then
17 grouped together (act 208) and the red pixel groups are filtered based on a set of
18 rules including geometric shape (act 210). The groups remaining after filtering in
19 act 210 are the candidate red-eye regions, and a multi-scale eye confirmation
20 process is used to identify whether each of the candidate red-eye regions is
21 actually part of an eye (act 212). An indication is then output of each of the
22 candidate red-eye regions, if any, identified as being part of an eye (act 214).

23 Fig. 5 illustrates an example of a suitable operating environment in which
24 the invention may be implemented. The illustrated operating environment is only
25 one example of a suitable operating environment and is not intended to suggest

1 any limitation as to the scope of use or functionality of the invention. Other well
2 known computing systems, environments, and/or configurations that may be
3 suitable for use with the invention include, but are not limited to, personal
4 computers, server computers, hand-held or laptop devices, multiprocessor systems,
5 microprocessor-based systems, programmable consumer electronics (e.g., digital
6 video recorders), gaming consoles, cellular telephones, image capture devices
7 (e.g., digital cameras), network PCs, minicomputers, mainframe computers,
8 distributed computing environments that include any of the above systems or
9 devices, and the like.

10 Fig. 5 shows a general example of a computer 242 that can be used in
11 accordance with certain embodiments of the invention. Computer 242 is shown as
12 an example of a computer in which various embodiments of the invention can be
13 practiced, and can be, for example, a computer 106 of Fig. 1, or a computer in
14 which system 120 of Fig. 2 is implemented. Computer 242 is illustrated as only
15 an example of a computing device that may be used with the invention; other
16 devices may alternatively used that include more components or alternatively
17 fewer components than those illustrated in Fig. 5. For example, a digital camera
18 may include only a processing unit, bus, system memory, universal serial bus
19 interface, and removable storage device.

20 Computer 242 includes one or more processors or processing units 244, a
21 system memory 246, and a bus 248 that couples various system components
22 including the system memory 246 to processors 244. The bus 248 represents one
23 or more of any of several types of bus structures, including a memory bus or
24 memory controller, a peripheral bus, an accelerated graphics port, and a processor
25 or local bus using any of a variety of bus architectures. The system memory 246

1 includes read only memory (ROM) 250 and random access memory (RAM) 252.
2 A basic input/output system (BIOS) 254, containing the basic routines that help to
3 transfer information between elements within computer 242, such as during start-
4 up, is stored in ROM 250.

5 Computer 242 further includes a hard disk drive 256 for reading from and
6 writing to a hard disk, not shown, connected to bus 248 via a hard disk drive
7 interface 257 (e.g., a SCSI, ATA, or other type of interface); a magnetic disk drive
8 258 for reading from and writing to a removable magnetic disk 260, connected to
9 bus 248 via a magnetic disk drive interface 261; and an optical disk drive 262 for
10 reading from and/or writing to a removable optical disk 264 such as a CD ROM,
11 DVD, or other optical media, connected to bus 248 via an optical drive interface
12 265. The drives and their associated computer-readable media provide nonvolatile
13 storage of computer readable instructions, data structures, program modules and
14 other data for computer 242. Although the exemplary environment described
15 herein employs a hard disk, a removable magnetic disk 260 and a removable
16 optical disk 264, it will be appreciated by those skilled in the art that other types of
17 computer readable media which can store data that is accessible by a computer,
18 such as magnetic cassettes, flash memory cards, random access memories
19 (RAMs), read only memories (ROM), and the like, may also be used in the
20 exemplary operating environment.

21 A number of program modules may be stored on the hard disk, magnetic
22 disk 260, optical disk 264, ROM 250, or RAM 252, including an operating system
23 270, one or more application programs 272, other program modules 274, and
24 program data 276. A user may enter commands and information into computer
25 242 through input devices such as keyboard 278 and pointing device 280. Other

1 input devices (not shown) may include a microphone, joystick, game pad, satellite
2 dish, scanner, or the like. These and other input devices are connected to the
3 processing unit 244 through an interface 268 that is coupled to the system bus
4 (e.g., a serial port interface, a parallel port interface, a universal serial bus (USB)
5 interface, an IEEE 1394 (Firewire) interface, etc.). A monitor 284 or other type of
6 display device is also connected to the system bus 248 via an interface, such as a
7 video adapter 286. In addition to the monitor, personal computers typically
8 include other peripheral output devices (not shown) such as speakers and printers.

9 Computer 242 operates in a networked environment using logical
10 connections to one or more remote computers, such as a remote computer 288.
11 The remote computer 288 may be another personal computer, a server, a router, a
12 network PC, a peer device or other common network node, and typically includes
13 many or all of the elements described above relative to computer 242, although
14 only a memory storage device 290 has been illustrated in Fig. 5. The logical
15 connections depicted in Fig. 5 include a local area network (LAN) 292 and a wide
16 area network (WAN) 294. Such networking environments are commonplace in
17 offices, enterprise-wide computer networks, intranets, and the Internet. In certain
18 embodiments of the invention, computer 242 executes an Internet Web browser
19 program (which may optionally be integrated into the operating system 270) such
20 as the "Internet Explorer" Web browser manufactured and distributed by Microsoft
21 Corporation of Redmond, Washington.

22 When used in a LAN networking environment, computer 242 is connected
23 to the local network 292 through a network interface or adapter 296. When used
24 in a WAN networking environment, computer 242 typically includes a modem 298
25 or other means for establishing communications over the wide area network 294,

1 such as the Internet. The modem 298, which may be internal or external, is
2 connected to the system bus 248 via a serial port interface 268. In a networked
3 environment, program modules depicted relative to the personal computer 242, or
4 portions thereof, may be stored in the remote memory storage device. It will be
5 appreciated that the network connections shown are exemplary and other means of
6 establishing a communications link between the computers may be used.

7 Computer 242 also includes a broadcast tuner 300. Broadcast tuner 300
8 receives broadcast signals either directly (e.g., analog or digital cable
9 transmissions fed directly into tuner 300) or via a reception device (e.g., via
10 antenna or satellite dish).

11 Computer 242 typically includes at least some form of computer readable
12 media. Computer readable media can be any available media that can be accessed
13 by computer 242. By way of example, and not limitation, computer readable
14 media may comprise computer storage media and communication media.
15 Computer storage media includes volatile and nonvolatile, removable and non-
16 removable media implemented in any method or technology for storage of
17 information such as computer readable instructions, data structures, program
18 modules or other data. Computer storage media includes, but is not limited to,
19 RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM,
20 digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic
21 tape, magnetic disk storage or other magnetic storage devices, or any other media
22 which can be used to store the desired information and which can be accessed by
23 computer 242. Communication media typically embodies computer readable
24 instructions, data structures, program modules or other data in a modulated data
25 signal such as a carrier wave or other transport mechanism and includes any

1 information delivery media. The term "modulated data signal" means a signal that
2 has one or more of its characteristics set or changed in such a manner as to encode
3 information in the signal. By way of example, and not limitation, communication
4 media includes wired media such as wired network or direct-wired connection,
5 and wireless media such as acoustic, RF, infrared and other wireless media.
6 Combinations of any of the above should also be included within the scope of
7 computer readable media.

8 The invention has been described in part in the general context of
9 computer-executable instructions, such as program modules, executed by one or
10 more computers or other devices. Generally, program modules include routines,
11 programs, objects, components, data structures, etc. that perform particular tasks
12 or implement particular abstract data types. Typically the functionality of the
13 program modules may be combined or distributed as desired in various
14 embodiments.

15 For purposes of illustration, programs and other executable program
16 components such as the operating system are illustrated herein as discrete blocks,
17 although it is recognized that such programs and components reside at various
18 times in different storage components of the computer, and are executed by the
19 data processor(s) of the computer.

20 Alternatively, the invention may be implemented in hardware or a
21 combination of hardware, software, and/or firmware. For example, one or more
22 application specific integrated circuits (ASICs) could be designed or programmed
23 to carry out the invention.
24
25

1 **Conclusion**

2 Although the description above uses language that is specific to structural
3 features and/or methodological acts, it is to be understood that the invention
4 defined in the appended claims is not limited to the specific features or acts
5 described. Rather, the specific features and acts are disclosed as exemplary forms
6 of implementing the invention.

7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25